A White-Box Approach to Function-Parameter Co-Estimation for Kinetic Models of Biochemical Networks

Calvin Hopkins

Senior Honors Thesis

Department of Electrical and computer Engineering

Tufts University

May 2012

Acknowledgements

Thanks to my advisor, Prof. Soha Hassoun, for her ideas, words of wisdom, and all other help through the entire project. Additionally thanks to Ehsan Ullah for helping generate and structure simulation data. I'd also like to thank Prof. Shuchin Aeron for being part of my defense committee and making sure I actually knew what I was talking about.

Dedication

To my mother
Who is always there to push me forward no matter how hard I resist

Abstract

Kinetic modeling attempts to describe the time-dependent behavior of every enzyme-catalyzed reaction in the network. Systems of ordinary differential equations express each reaction rate as a function of metabolite concentrations and rate constant parameters. The mechanistic knowledge of enzyme kinetics however is not always available. Furthermore, estimating a consistent set of rate parameters from time-series data requires a large experimental effort for even a moderately-sized network. This work develops approximate kinetic expressions for a module (*connected groups of reactions*) instead of developing kinetic expressions for *each* reaction within the module. Our modeling utilizes Convenience Kinetics to capture system dynamics. We develop a genetic algorithm to explore the trade offs between expression accuracy and simplicity, in terms of including variables representing the concentration of metabolites internal to the module. When eliminating an internal variable, our algorithm compensates by calculating a new set of parameter values and shifting the roles of remaining metabolites as activators and inhibitors. Our validation efforts on three test cases demonstrate such tradeoffs, and show that modest loss of accuracy is possible when some internal metabolite concentrations are eliminated and when the resulting system is gamed to compensate for the missing variables.

1. Introduction

Understanding and engineering the dynamic behavior of living cells promise to propel innovations in biotechnology and health care. Computational models that can capture the complex, time-dependent behavior of a cell should facilitate the design and optimization of many industrially relevant microorganisms. Preferred over constraint-based models, kinetic models describe the time-dependent behavior of every enzyme-catalyzed reaction in the network based on mechanistic knowledge. Systems of ordinary differential equations (ODEs) or their transforms are written for each reaction as a mathematical (reaction rate law) function of metabolite concentrations and rate constant parameters. Kinetic model construction, however, is hindered by the limited availability of kinetic data (mechanistic expressions and rate constant parameters) especially when considering a genome-scale model (Costa, Machado et al. 2011). Published kinetic data have limitations and inconsistencies as reaction mechanisms remain unknown or poorly documented within databases, and experimental conditions under which kinetic parameters have been determined are unavailable. Newly available modern high-throughput techniques reduce large experimental efforts and facilitate collecting consistent sets of time-series data. The challenge then becomes utilizing this data to solve the "inverse" problem: estimation of model parameters and the identification of the structure and regulation of the underlying biological networks based on time-series data.

To address the challenges in solving the inverse problem, several modeling and computational efforts have been explored (Chou and Voit 2009). Modeling has focused on using 'canonical' nonlinear models, where structure is fixed and individuality comes from parameter values. Example systems include the S-System (Voit 2000), Generalized Mass Action formalism(Peschel and Mende 1986), and linlog approximations (Hatzimanikatis and Bailey 1996) (Visser and Heijnen 2002). These models behave similarly if system variables do not deviate from selected operating states. Canonical models have several advantages: capturing non-linearity, capitalizing on known stoichiometric structures, and eliminating dependence on mechanistic approaches that are either unknown or not appropriately discerned considering the accuracy of the time-series data. Additionally, canonical models simplify the mathematical formulations for parameter estimation (identification).

When using canonical models, the inverse problem is cast as a non-linear optimization. Given a set of ODEs, the objective function is to minimize the error between calculated and measured dependent variables based on a set of parameter choices. The inputs to the problem are the measured initial values of the independent variables (e.g. metabolite concentrations) and the stoichiometric structure of the underlying biological network. As with other non-linear optimization problems, guaranteeing a globally optimal solution is exceedingly difficult. Computational efforts mainly utilize two approaches: stochastic search and gradient-based approaches. Stochastic approaches utilize genetic algorithms and genetic programming, while others use meta-heuristics such as simulated annealing (Gonzalez, Kuper et al. 2007), ant colony optimization (Zuñiga, Pasia et al. 2008), and particle swarm optimization (Naval, Sison et al. 2006). These approaches are characterized by simultaneous exploration and stochastic refinement of many solutions, except for simulated annealing which allows for restarting strategies. Gradient-based local methods are dependent on initial conditions and often yield local minimum despite multiple restart strategies as the solution space is typically non-convex. It is generally agreed that global methods, while computationally expensive, are more appropriate

(Moles, Mendes et al. 2003). It is also recommended that a suite of diverse computational methods should be available as not one performs best for all problems (Mendes and Kell 1998). Additional computational efforts focused on algorithmic support to speed up integration, the key computational bottleneck during parameter estimation, and smoothing noisy data.

With the rise of genome scale models and the desire to model whole cell dynamics, there is a need to simplify (reduce) dynamic models yet maintain accuracy and biochemically meaningful model parameters. Earlier efforts in model reduction focused on three techniques (Okino and Mavrovouniotis 1998): lumping of reactions based on intuition of very fast or very slow reactions, sensitivity analysis to identify and then eliminate parts of the model that are unimportant to the properties of interest, and timescale-based methods that eliminate parts of the model operating at a different time scale than the one of interest. Splitting a biochemical network into a subsystem and a surrounding environment, and linearizing the behavior of the environment around a steady state enables using a reduced set of variables to describe the environment's dynamic modes that dominate its interaction with the subsystem(Liebermeister, Baur et al. 2005). This method retains the detailed modeling of the subsystem while approximating the behavior of the environment. Another method, ENVA (Elimination of Nonessential Variables), explores the possibilities of eliminating variables (metabolite concentrations) by holding them at their steady-state concentration and evaluating the impact on the systems' dynamics (Dano, Madsen et al. 2006). Classifying the turn-over rate of metabolite pools as fast or slow and using linlog kinetics to express linear relations between fast and slow metabolites, a reduced model with lumped reactions is constructed and its parameters can be identified using established parameter estimation technique (Nikerel, van Winden et al. 2009).

In this study, we investigate a novel model reduction technique that develops approximate kinetic expressions for a module (*connected groups of reactions*). Like prior model reduction work, we seek to simplify the model. However, our interest is hiding module details that do not affect the external behavior of the system. The usefulness of this approach is in building kinetic expressions of modules that are the result of modularity analysis (Sridharan, Hassoun et al. 2011), when the emphasis is on the system behavior and not the module details. This type of analysis is dominant across all Engineering disciplines, and is referred to as *abstraction.* Instead of modeling every reaction in the network, in our approach we will iteratively combine reactions generating new less complex models and using Convenience Kinetics (Liebermeister and Klipp 2006) to identify a number of mathematical expressions which govern the behavior of the network. The resulting number of mathematical expressions is smaller compared to the number of equations needed to describe every part of the original network. This systematic elimination of nodes reduces the total number of equations and parameters needed to be co-estimated through our multi-scale modeling technique. Our work utilizes Convenience Kinetics, which offers a systematic way of generate rate equations. Convenience Kinetics support plausible biological properties including enzyme saturation, regulation by activators and inhibitors, and allowing reversible and irreversible reactions. Furthermore, each rate equation can be specified by a small number of parameters. Our approach is applied to three example systems, where data sets are used to both train and validate the models.

2. Methods

2. 1. Review of Convenience Kinetics

Our approach utilizes a "convenience kinetics" which encompasses all possible stoichiometries as well as the effect of activators and inhibits on specific enzymes (Liebermeister and Klipp, 2006). Convenience kinetics limits the set of tunable parameters to the substrate and product constants and the turnover rates. The generalized formula is shown in Equation 1 for reversible reactions.

$$v_l = E_l \prod_m \overbrace{h_A(c_m, k_{lm}^A)^{w_{lm}^+}}^{Activator\ Pre-Factors} \overbrace{h_I(c_m, k_{lm}^I)^{w_{lm}^-}}^{Inhibitor\ Pre-Factors} \times \frac{\overbrace{k_{+l}^{cat} \prod_i \tilde{c}_{li}^{n_{li}^-}}^{Reactant\ Metabolites} - \overbrace{k_{-l}^{cat} \prod_i \tilde{c}_{li}^{n_{li}^+}}^{Product\ Metabolites}}{\prod_i \sum_{m=0}^{n_{il}^-} (\tilde{c}_{li})^m + \prod_i \sum_{m=0}^{n_{il}^+} (\tilde{c}_{li})^m - 1}$$

$$\tilde{c}_{li} = {}^{c_i}/_{k_{li}^M} \text{ and } \tilde{k}_{li}^M = k_{li}^M k_i^G$$

Equation 1: Convenience Kinetics General Form for Reversible Reactions

E_l need not be independent; therefore, the turnover rates, k_{+l}^{cat} and k_{-l}^{cat}, from M-M kinetics, can be modified to account for the enzyme concentration. The M-M constants, k^M, have been replaced with substrate constants $k_{a_i}^M$ and product constants $k_{b_i}^M$. Where the Michaelis-Menten cannot model enzyme regulation through activators and inhibitors, convenience kinetics uses the activator and inhibitor pre-factors for this process. The expanded representations for these pre-factors are shown below in Equation 2.

$$h_A(c_m, k_{lm}^A)^{w_{lm}^+} = 1 + \frac{c_m}{k_{lm}^A} \text{ and } h_I(c_m, k_{lm}^I)^{w_{lm}^-} = \frac{k_{lm}^I}{k_{lm}^I + c_m}$$

(Liebermeister and Kilpp, 2006.)

Equation 2: Expanded Pre-Factors for Convenience Kinetics

For irreversible reactions, a separate form of Convenience Kinetics is used. This is derived from the original form when, as in a reversible reaction, the product constants approach infinity. The form for this reaction is below in Equation 3.

$$v_l = E_l \prod_m \overbrace{h_A(c_m, k_{lm}^A)^{w_{lm}^+}}^{Activator\ Pre-Factors} \overbrace{h_I(c_m, k_{lm}^I)^{w_{lm}^-}}^{Inhibitor\ Pre-Factors} \times \frac{\overbrace{k_{+l}^{cat} \prod_i \tilde{c}_{li}^{n_{li}^-}}^{Reactant\ Metabolites}}{\prod_i \sum_{m=0}^{n_{il}^-} (\tilde{c}_{li})^m}$$

Equation 3: Convenience Kinetics General Form for Irreversible Reactions

Algorithm Pseudocode

- Generate Initial Solution using all internal metabolites from predetermined stoichiometric structure of the network and add to solution set with same number of metabolites
- While current solution set includes internal metabolites
    - For each solution in set
        - Create set of random starting point for constants and activator inhibitors called the population (population size: 200)
        - While generations++ != 250
            - Evaluate All solutions
            - Rank by ascending Error Values
            - Create a new population by the following methods
                - Keep top 20% (Elitism Criteria)
                - Remove bottom 20%
                - Top 80% move on either through crossover or mutation
                    - Mutation Rate: 20%
                    - Crossover Rate:80%
                    - Mutation accomplished by re-randomizing parameters to keep diversity due to strong elitism
            - Remove old population, repeat with new
        - Sort final solutions by error value and store best solution as a potential best solution for the set including solutions with this number of metabolites
    - Sort each solution in the best solutions set by error value
    - If we can eliminate another metabolite
        - For top 20% of solutions in the current set, add all possible solutions generable by removing 1 more metabolite to the next set
        - For the rest of the solutions crossover 40% of possible solutions to increase the search through the solution space and to increase diversity
    - Proceed to next set of solutions (ones with 1 fewer metabolite)

Code 1: Pseudo-code for our Algorithm

2.2. Algorithm

Our algorithm uses a systematic approach of eliminating metabolites and comparing similar solutions to develop a comprehensive set of solutions with which we can analyze tradeoffs between model accuracy and model complexity (number of internals used in the model).

The error metric used is the normalized root-mean-square deviation (NRMSD) error between measured data and the data calculated by that specific model. The advantage of using the NRMSD is that it takes into account the possible range of the values. This metric provides a more accurate error calculation than that of the RMSD. The formula used for NRMSD is given in Equation 4.

$$NRMSD = \frac{\sum_{i:External\ Metabolites} \sqrt[2]{\frac{\sum_{t:time}\left(\frac{measured_i[t] - calcualted_i[t]}{max_i - min_i}\right)^2}{(number\ of\ measurements)}}}{(num\ Externals)}$$

Equation 4: NRMSD Calculation

The individual physical reactions modeled using Convenience Kinetics to capture reaction-product relationships comprise our initial system model. These initial reactions do not have activators or inhibits associated with them; these characteristics are discovered by the algorithm. This model is then added to a set of models all of which include the same number of internal metabolites. For the first and last solutions created, corresponding to models using all and none of the internal metabolites, these sets contain only one model. The algorithm then commences with this initial set of models containing solely the model for which all internal metabolites are included.

For each model in the current set, our method attempts to determine the values of parameters used by convenience kinetics as well as which metabolites are used as activators and inhibitors for each equation in the model. This process is accomplished by a genetic algorithm (GA) which is run on each solution. This GA implements a few proven methods to improve diversity of the population of solutions and reduce the convergence factor. A random initial population is created by selecting random values within a predetermined range for all parameters, and by randomly selecting metabolites to be used as activators and inhibitors for each equation. Additionally, our GA implements an elitism criterion, which maintains a percentage of the fittest solutions from each generation for subsequent generations to keep the quality of solutions larger. Although GAs often require long runtimes to properly generate accurate results, the combination of this with a high mutation rate decrease the necessary runtime in addition to improving the result characteristics (Laumanns, Zitzler et al. 2001). Mutation is implemented by randomizing the constants and the activator and inhibitor characteristics for a particular solution.

In addition to mutation and elitism, crossover between two individuals of a specific generation is the only other method for creating solutions in subsequent generations. Our crossover mechanics are performed using lightly weighted tournament selection, in combination with anti-elitism, to improve runtime while still maintaining diversity. Tournament selection gives each element in the current population a relative weighting corresponding to how good it is compared to each other element in the set. Although this appears to enable and facilitate premature convergence, it maintains diversity enough to prevent that undesired characteristic (Miller and Goldberg 1995). Our tournament selection also implements an anti-elitism criterion where all solutions in the bottom 20% of the population had their crossover privileges removed. This method speeds up runtime of our GA and the quality of solutions generated.

Once this GA has finished running for each model in the current set, the best results are used to determine how to create the next generation of models. Additionally, these best results are used for final analysis to see how the error varies between different numbers of metabolites having been eliminated. Another GA type crossover and mutation strategy is performed on this set of best solutions to determine the next generation of models. The best results from each model in the current set are ordered by NRMSD error. A specific model's children are identified as having the same metabolites eliminated and then exactly one additional metabolite eliminated. For the top 20% of the current set of solutions, all children are generated by systematically eliminating one more metabolite for each child. This is the equivalent of elitism and mutation. The next 60% of solutions are bred amongst themselves with a crossover rate of 40% to generate a diverse population of new solutions. The bottom 20%, based by NRMSD, of the current set of models does not exist in any form in the next generation. This method maintains diversity while consolidating the search through the total solution space.

This process is repeated until the current set of models contains only one solution, the one

with all internals eliminated. This is the termination point of our algorithm.

The final result for our algorithm is a set of best solutions, minimum NRMSD error, for models that include specific numbers of internal metabolites ranging from all metabolites to no metabolites. These results can be mapped as a pareto-optimal front comparing error and number of internals used (simplicity of solution).

3. Results

In this work, we use three example bio-chemical networks from the literature with varying degrees of complexities. We used the published models (equations and parameters) to generate both tuning and validation data. All computations were performed on Redhat Enterprise Linux using a 4 core machine. The algorithm is implemented in C++ and uses the Boost libraries for the Euler Stepping function. Each system is tuned and tested over multiple sets of data, 50% of which are used for tuning, and 50% for validation. The initial metabolite concentrations were randomly chosen from within a range of values appropriate for each system.

3.1 Case 1: An *In vitro* Multi-Enzyme System

The first network is an *in vitro* multi-enzyme system of an *E. coli* strain AG1 from the ASKA clone library (Ishii, Suga et al. 2007). The system was specified using three equations and is shown in Figure 1. The time-series data for these reactions was generated using the kinetic rate functions and their corresponding parameters and ODEs detailed in (Ishii, Suga et al. 2007). We used 5 tuning data sets and 5 validation data sets.

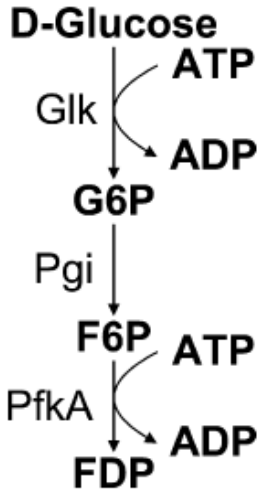


Figure 1: *E. Coli* AG1 Network (Ishii, Suga et al. 2007)

The algorithm discovers four solutions: The first includes all internal metabolites. The second and third solutions eliminate a single internal metabolite (G6P and F6P in turn), and the fourth eliminates all internals.

Figure 2 illustrates the each of the external metabolite's concentrations over time for the best solutions at every possible number of internal metabolites for the tuning data used for our algorithm.

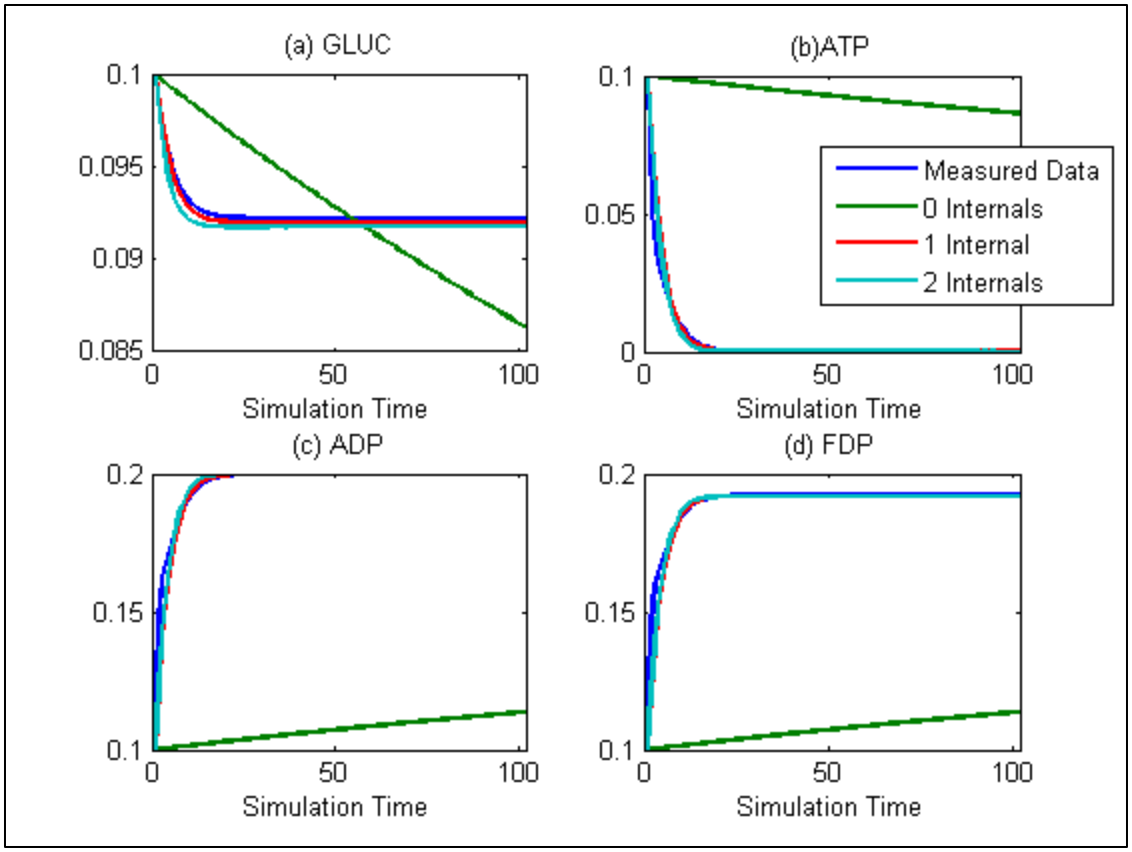


Figure 2: Metabolite Concentrations for Case 1 Externals (Tuning Data)

Table 1 presents the number of times each metabolite appears as activator, inhibitor across all four generated solutions sets.

Names	*GLUC*	*G6P*	*F6P*	*FPD*	*ATP*	*ADP*
Activators	1	0	2	3	1	2
Inhibitors	1	0	1	0	0	2

Table 1: Activators and Inhibitors for Case 1

Figure 3 is a summary for the NRMSD Errors for the set of boundary metabolites through different solutions. The extrema are shown for each subset of solutions characterized by having the same number of internal metabolites.

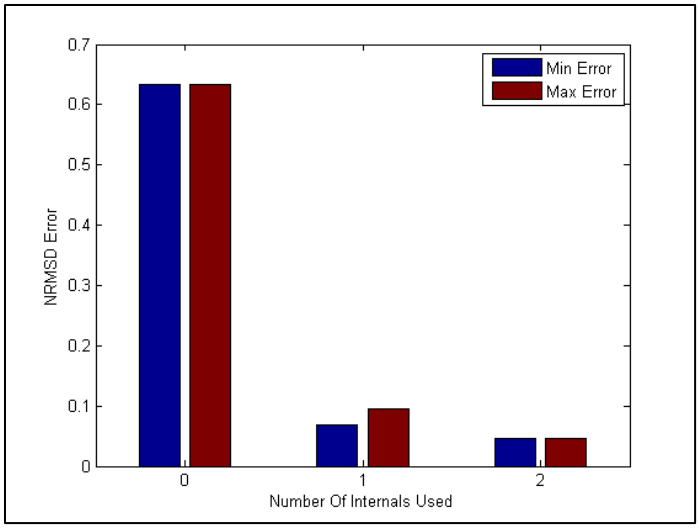


Figure 3: Error Limit Summary for Case 1 (Tuning Data)

The error increases when more internal metabolites are eliminated. The error value is 0.00196305 for all metabolites included vs. 0.0282421 for all eliminated. The values displayed show the calculated errors for the tuning data. The data presented show a distinct tradeoff between number of internals used and NRMSD values. The data corresponding to the minimum error forms a pareto-frontier to best demonstrate this tradeoff and the advantages of specific solutions.

Figure 4 is similar to Figure 2 in that it shows the concentrations of each of the external metabolites; however, the data used for these following graphs is generated by the *validation* data. Figure 5 summarizes the limits of the NRMSD errors generated for the validation data for the same solutions. Table 2 identifies which metabolites were used for specific solutions with the errors above.

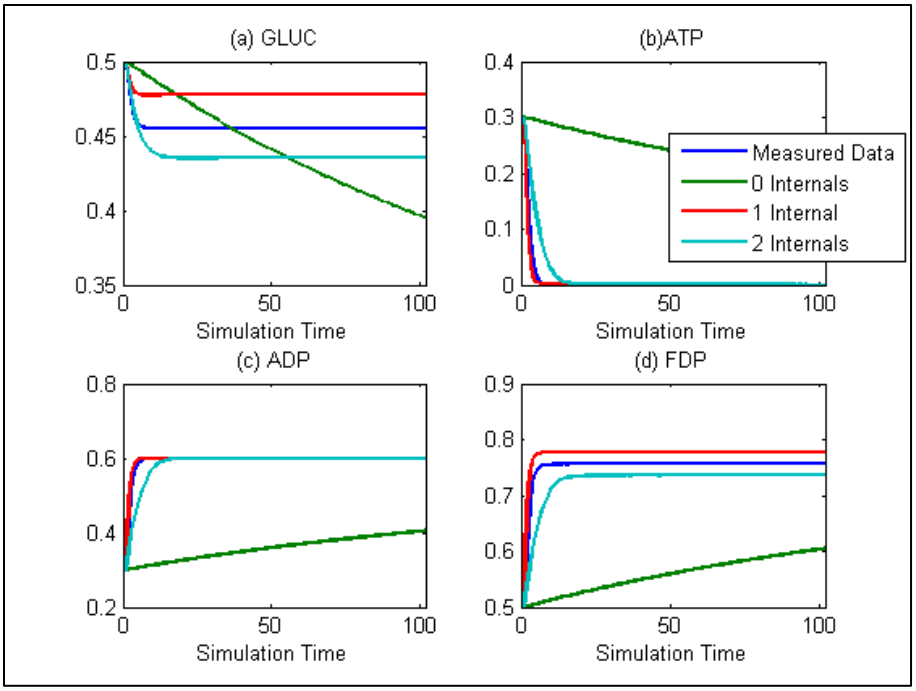


Figure 4: Metabolite Concentration for Case 1 Externals (Validation Data)

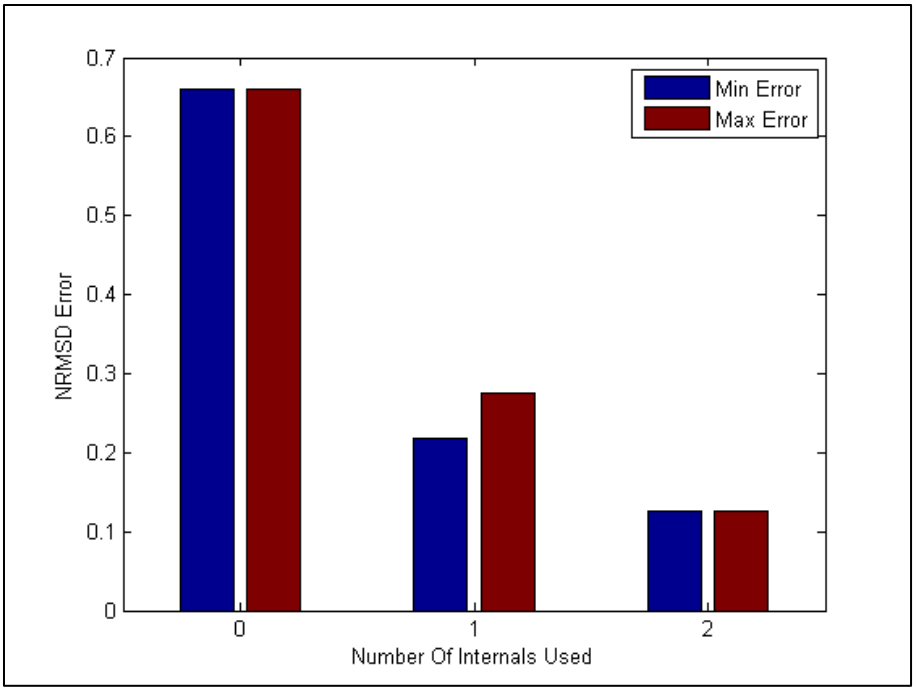


Figure 5: Error Limit Summary Case 1 (Validation Data)

	0 Internals	*1 Internal*	*2 Internals*
Min Error Solutions	None	G6P	All
Max Error Solution	None	F6P	All

Table 2: Metabolite Use for Solutions

3.2 Case 2: A Metabolic Pathway with Activators and Inhibitors

The system used for the second evaluative case is shown in Figure 6. This larger inverse problem has been previously used by Mendes as a parameter estimation problem implemented most efficiently through evolutionary programming (EP) (Mendes, 2001). We use the example here due to the similarities between EP and genetic programming (GP) in addition to the physical structure of the network allowing our method of Convenience Kinetics to properly model the system.

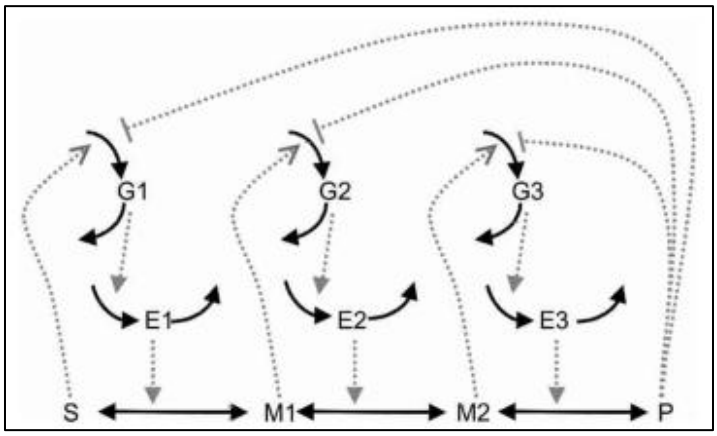


Figure 6: Case #2. Three-Step Pathway Network (Moles, Mendes et al. 2003)

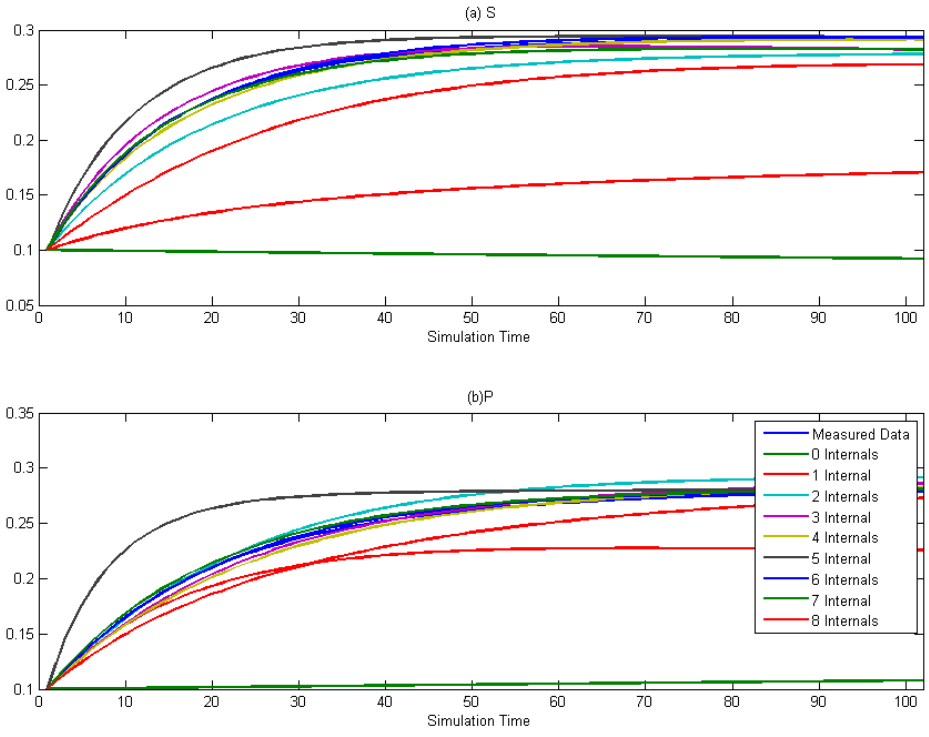


Figure 7 shows the concentrations of the external metabolites, S & P, from our tuning sets for all 8 best-solutions, each corresponding to a different number of eliminated metabolites. Table 3 tabulates the number of times that each metabolite has been used as an activator and inhibitor in

the best solutions for each number of metabolites. Figure 8 illustrates the pareto front for the tuning error distribution. This is the equivalent of Figure 3 for Case 2. Table 4 identifies for Figure 8 which internal metabolites were used for the min and max solutions. Figure 9 shows error distribution per metabolite.

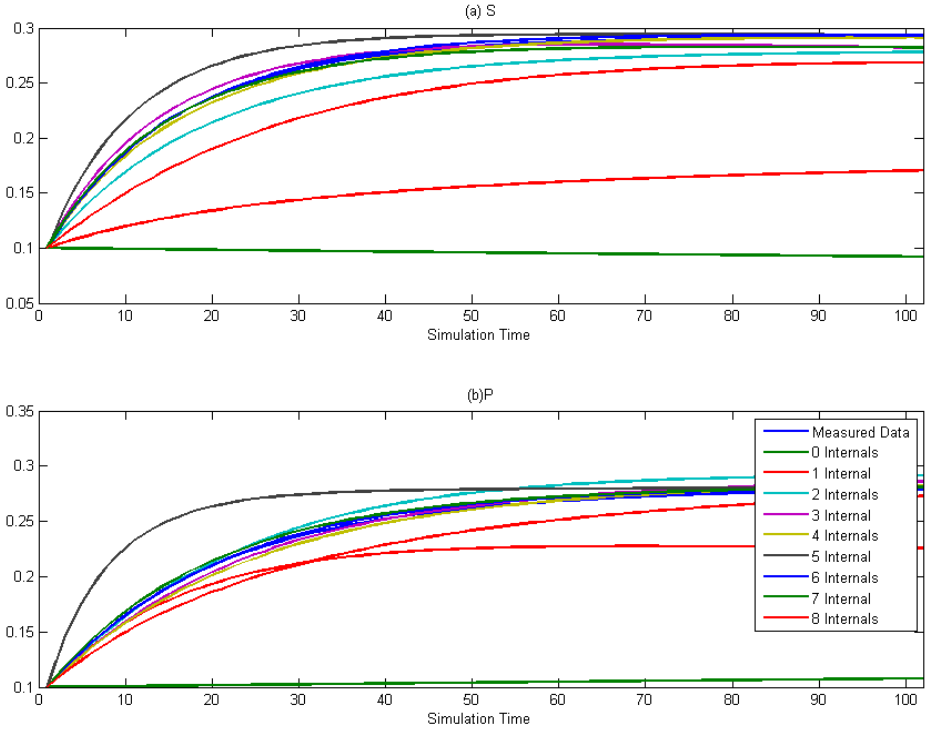


Figure 7: Metabolite Concentrations for Case 2 (Tuning Data)

Names	*P*	*S*	*G1*	*G2*	*G3*	*E1*	*E2*	*E3*	*M1*	*M2*
Activators	6	4	0	2	3	3	1	2	6	1
Inhibitors	3	5	1	1	2	3	4	4	5	4

Table 3: Activator and Inhibitors for Case 2

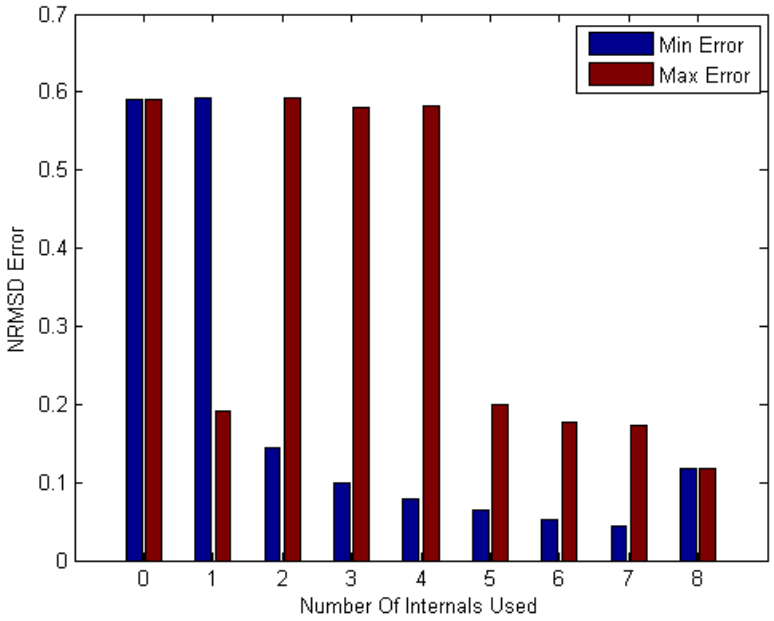


Figure 8: Error Limit Summary for Case 2 (Tuning Data)

	0 Internals	***1 Internal***	***2 Internals***	***3 Internals***	***4 Internals***	***5 Internals***	***6 Internals***	***7 Internals***	***8 Internals***
Min Error Solutions	None	M1	M1,M2	G3,M1,M2	G3,E1,M1, M2	E1,E2,E3, M1,M2	G2,E1,E2, E3,M1,M2	G2,G3,E1, E2,E3,M1, M2	All
Max Error Solutions	None	G2	G2,E2	G1,G2,G3	G2,G3,E1, E2	G2,G3,E1, E2,E3	G1,G2,G3, E1,E2,E3	G1,G2,G3, E1,E2,E3, M2	All

Table 4: Metabolites used for Min/Max Error Solutions for Case 2

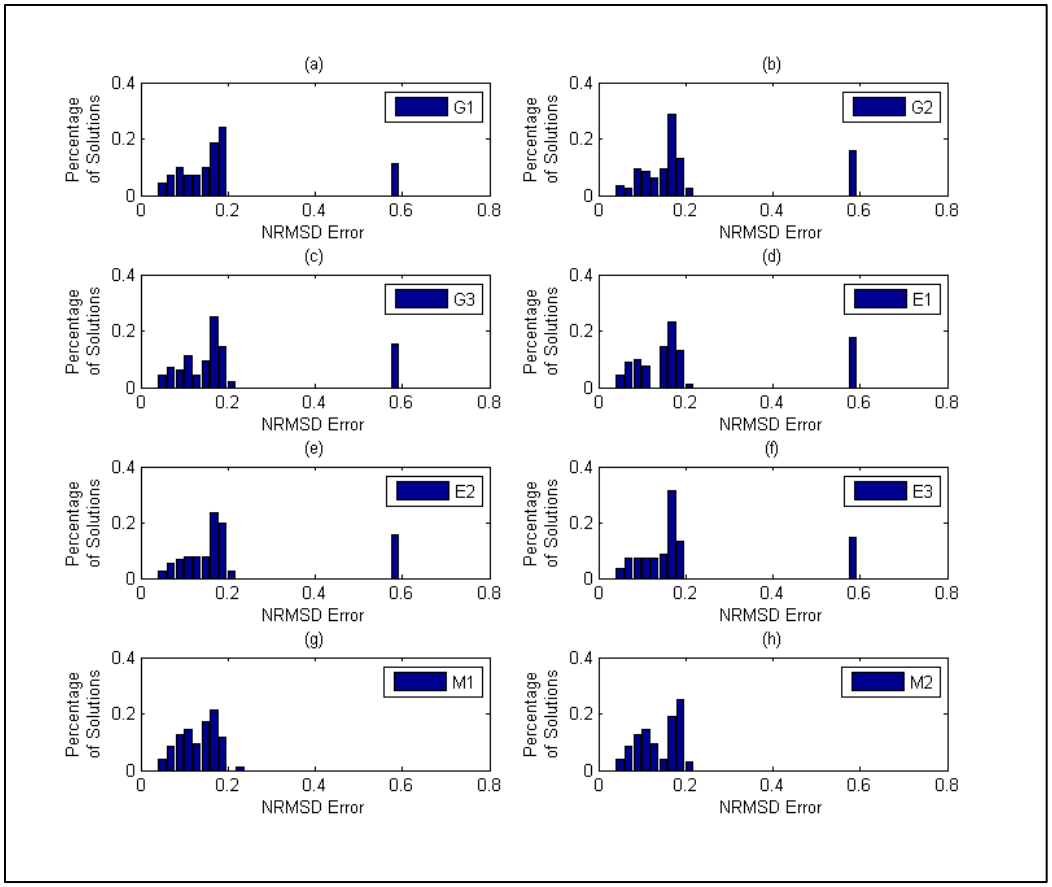


Figure 9: Error Distribution Per Metabolite

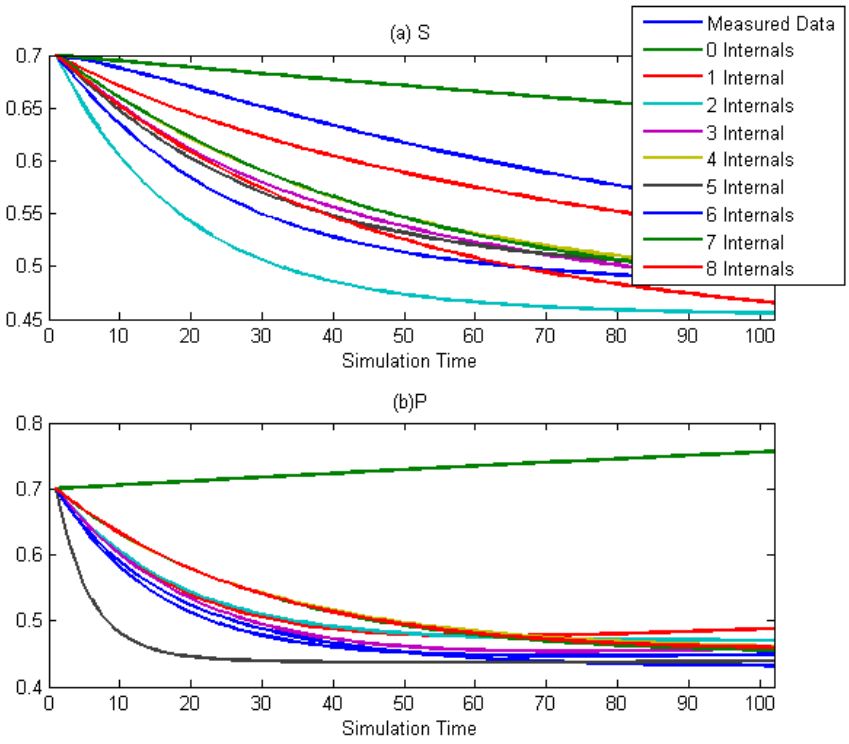


Figure 10: Metabolite Concentrations for Case 2 (Validation Data)

shows metabolite concentrations for the validation data. Figure 11 shows the associated error limits for the validation data.

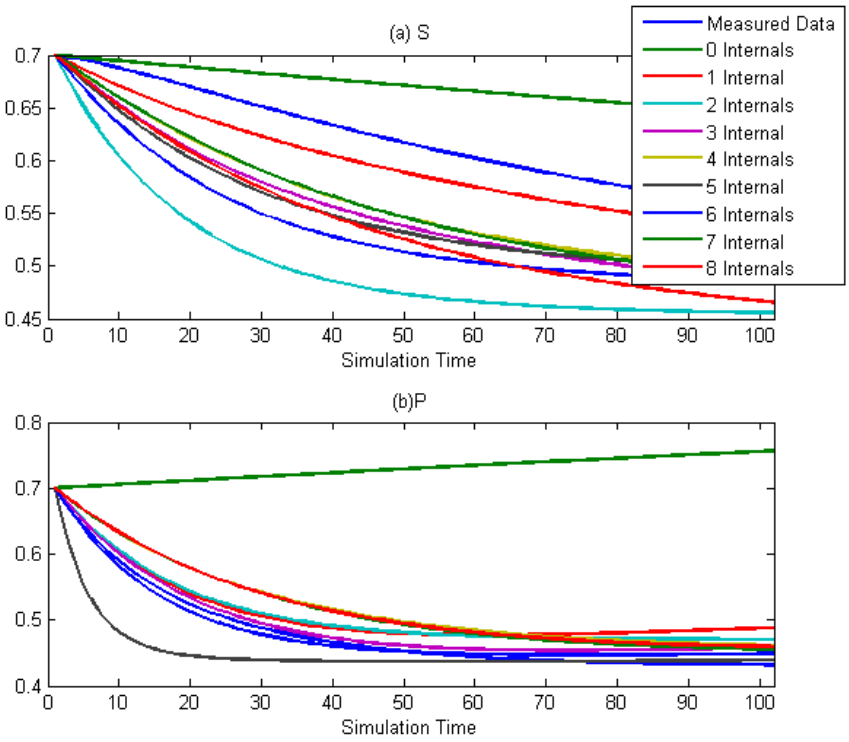


Figure 10: Metabolite Concentrations for Case 2 (Validation Data)

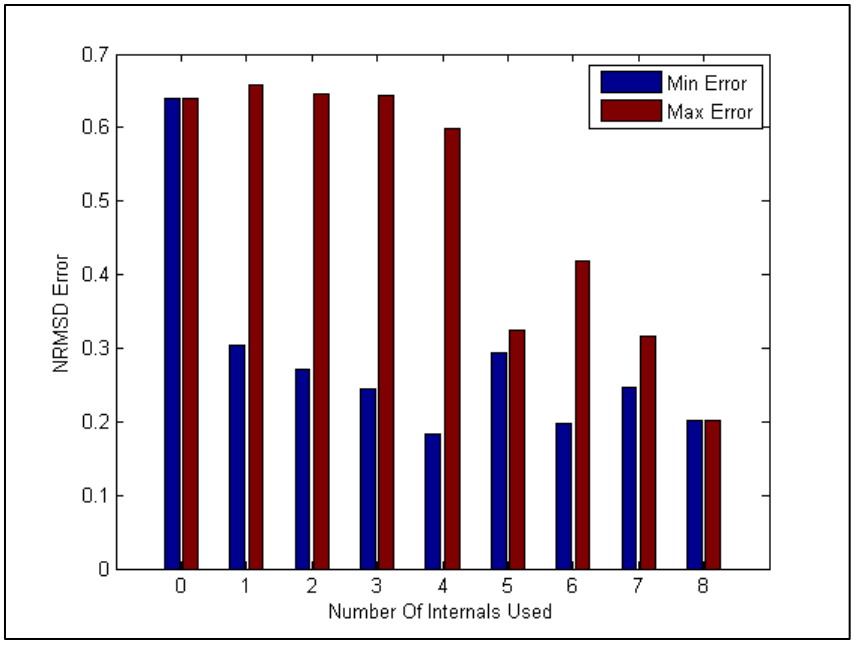


Figure 11: Error Limits for Case 2 (Validation Data)

The algorithm explores a large number of elimination, but does not exhaustively evaluate all 2^8 scenarios. However, the algorithm identifies one best solution for each number of eliminated metabolites from a subset of the possible solutions for that number of metabolites. Figure 12 shows the total # of solutions generated, and how many metabolites actually particpate in the each solution. The algorithm provides an upper limit of the metabolites to be included.

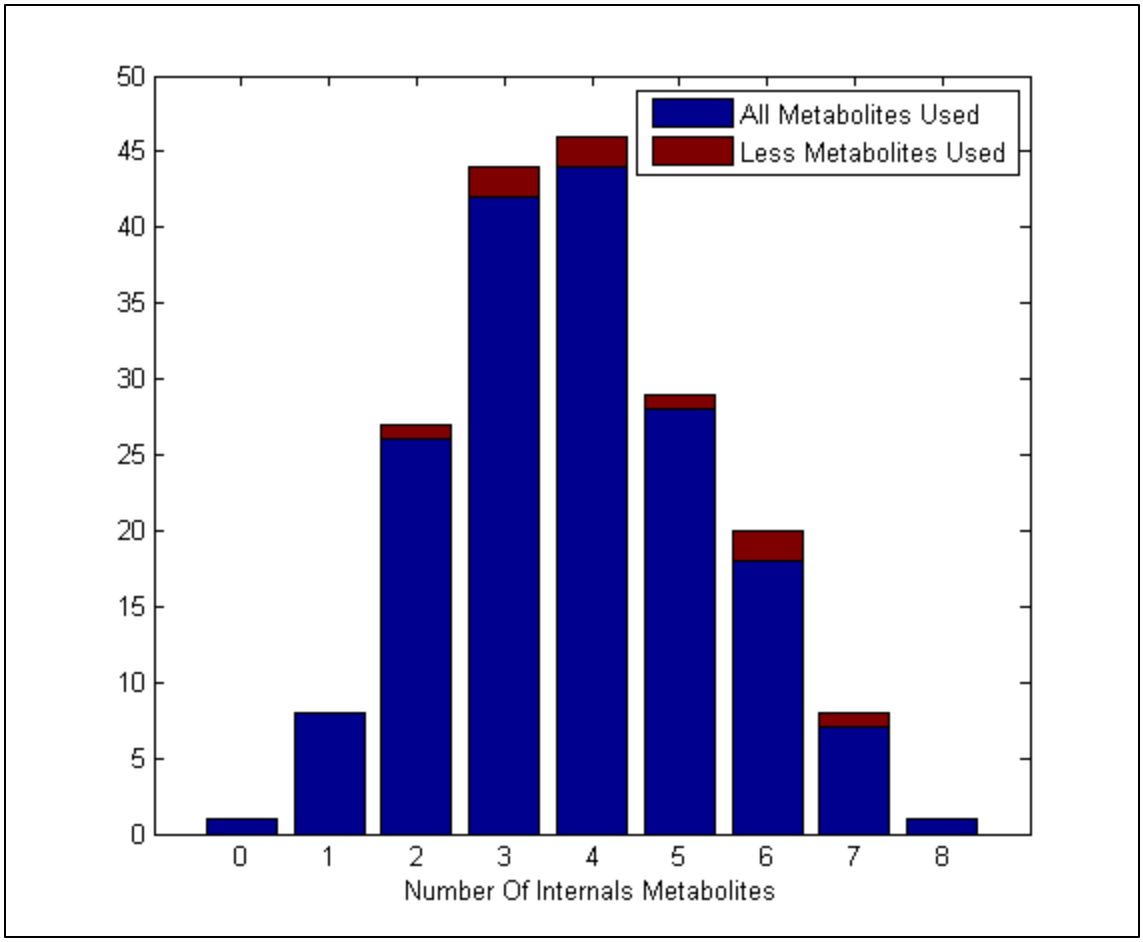
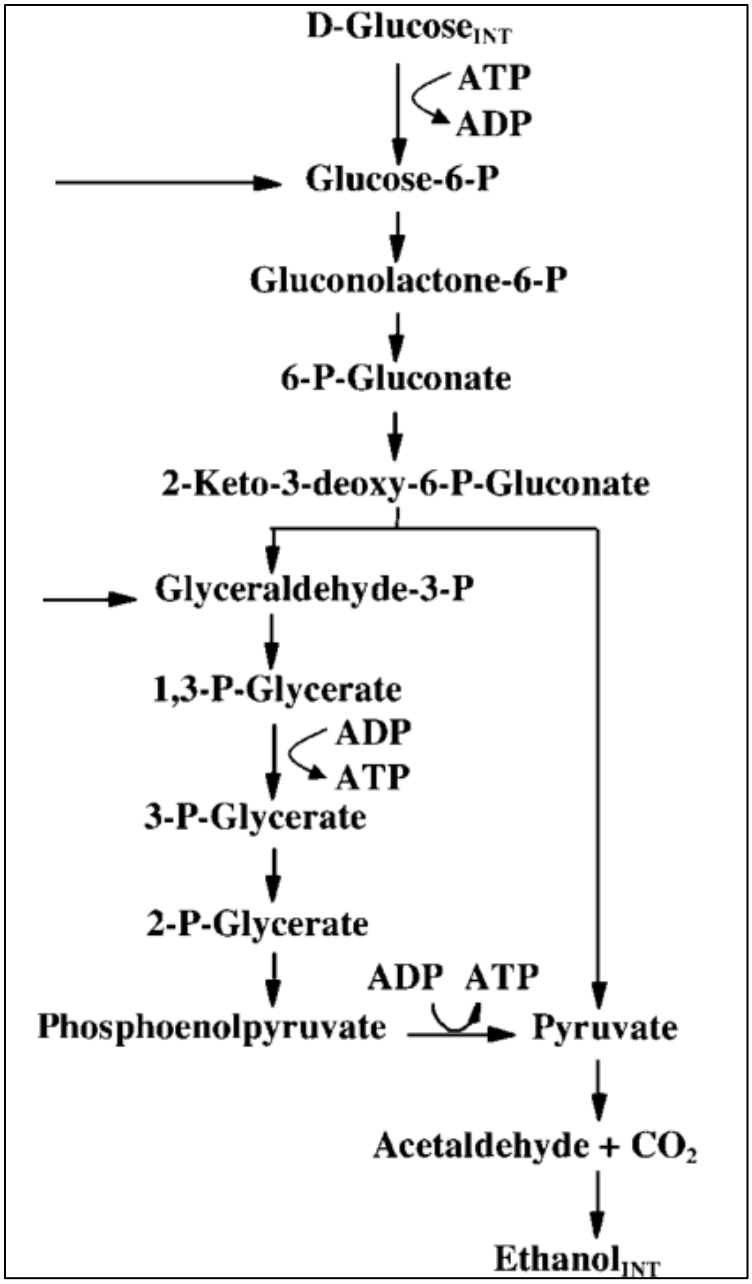


Figure 12: Number of Solutions Using Exactly Required Number of Metabolites

3.3 Case 3: Entner Doudoroff Pathway

The system used for our third case is shown in Figure 12 and is part of the Entner Doudoroff Pathway. Specifically, the network is a subsystem of *Zymomonas mobilis* (Altintas, Eddy et al. 2006). Because both Glucose-6-P (G6P) and Glyceraldehyde-3-P (GAP) both have a constant flux from an external source, they are required to be internal metabolites that cannot be eliminated. For this reason, the simplest solution for this network includes 2 internal metabolites. Additionally, to show the effect of graphical pathway compression on the algorithm, Gluconolactone-6-P, 6-P-Gluconate, 3-P-glycerate, 2-P-Glycerate, and Phosphoenolpryuvate have been manually eliminated as a starting condition for the network.



System 1:Physical Structure of Zymomonas mobilis(Altintas, Eddy et al. 2006)

Figure 13 shows the concentrations of the metabolites for specific solutions through our simulation. Bceause of constraints that are forced upon this network the fewest number of metabolites used internally is two for this example.

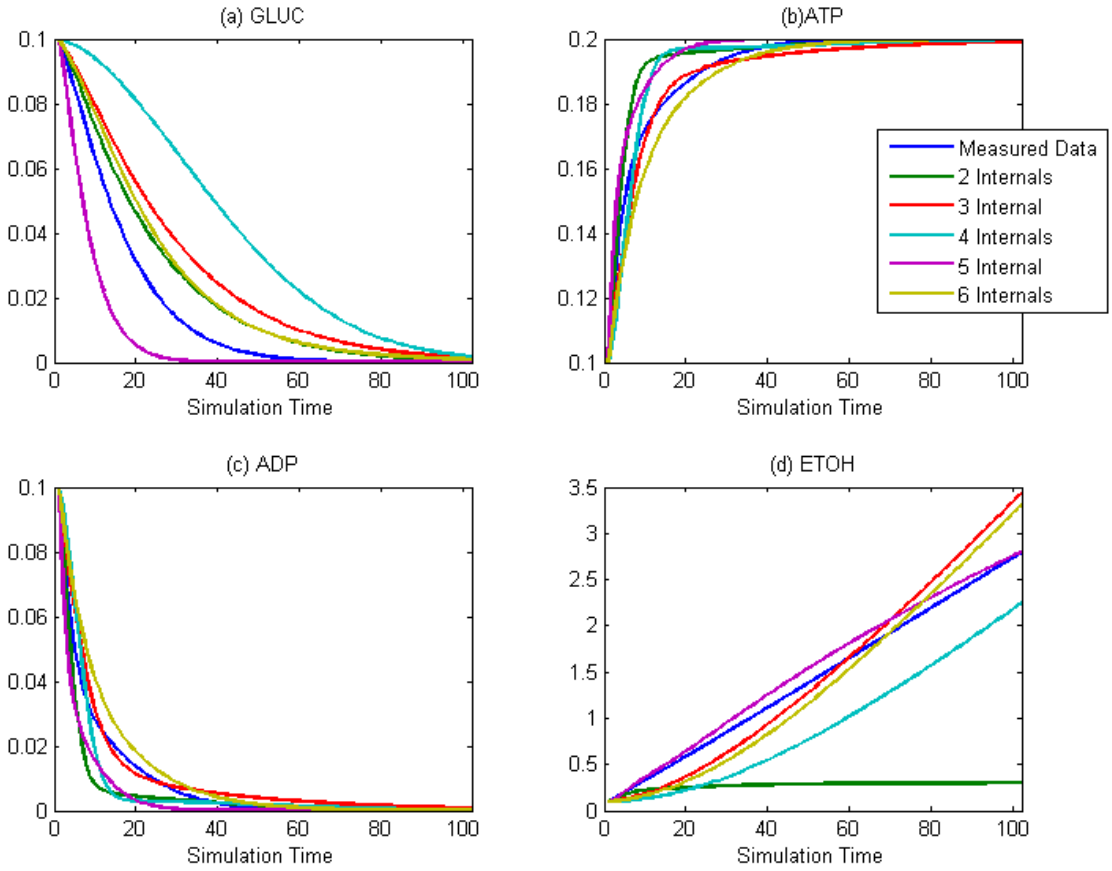


Figure 13: Metabolite Concentrations for Case 3 Externals (Tuning Data)

Table 5 shows the number of times each metabolite has been used as an activator or inhibitor for this system through our algorithm. Only metabolites not removed through pre-processing graph compression have been shown.

Names	*GLUC*	*GLUC6P*	*ATP*	*ADP*	*KDPG*	*G3P*	*GAP*	*PYR*	*BPG*	*ETOH*
Activators	9	5	5	3	3	3	5	0	4	6
Inhibitors	5	5	8	9	1	2	4	5	3	4

Table 5: Activator and Inhibitors for Case 3

Figure 14 shows the minimum and maximum errors for the solutions generated with the tuning data for case 3. This illustrates the limits of the errors for the tuning cases.

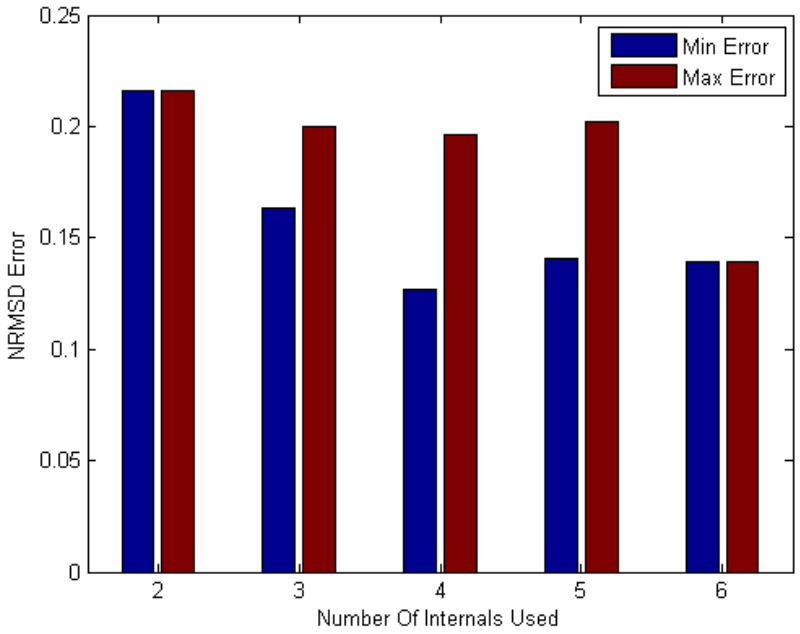


Figure 14: Error Limit Summary for Case 3 (Tuning Data)

Each solution appearing in Figure 14 has a corresponding set of internal metabolites that are used to details the physicallity of each reaction. The following table, Table 6, shows the internal metabolites used for each of the solutions above.

	2 Internals	***3 Internals***	***4 Internals***	***5 Internals***	***6 Internals***
Min Error Solutions	GLUC6P GAP	GLUC6P GAP PYR	GLUC6P GAP PYR BPG	GLUC6P KDPG GAP PYR BPG	GLUC6P KDPG G3P GAP PYR BPG
Max Error solutions	GLUC6P GAP	GLUC6P G3P GAP	GLUC6P KDPG G3P GAP	GLUC6P KDPG G3P GAP BPG	GLUC6P KDPG G3P GAP PYR BPG

Table 6: Metabolites Used for Min/Max Error Solutions for Case 3

Figure 15 shows the error distribution per metabolite. It identifies which metabolites are involved in solutions at certain error values.

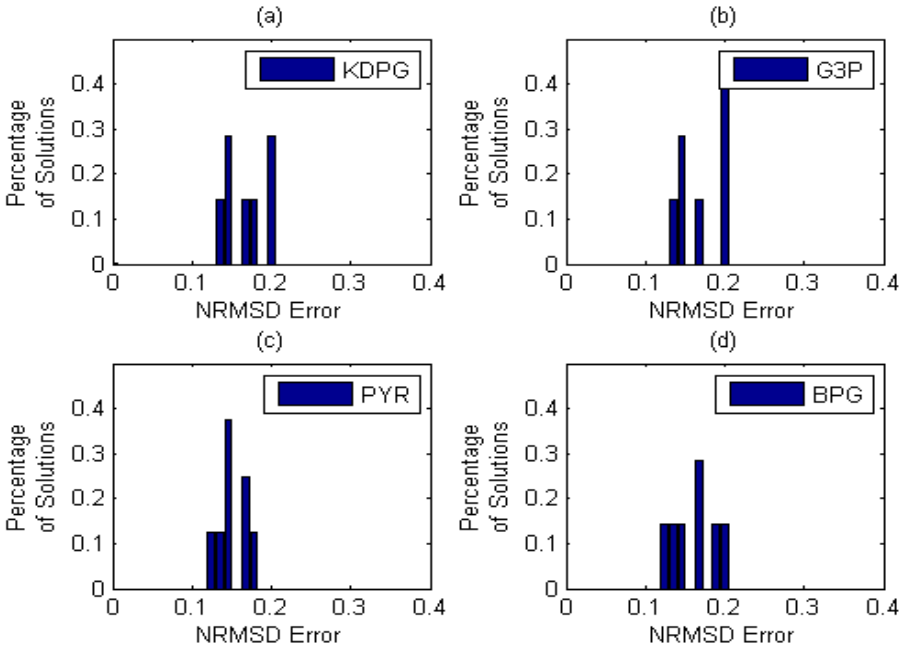


Figure 15: Error Distribution Per Metabolite

Figure 16 shows the concentrations over time for the validation data sets.

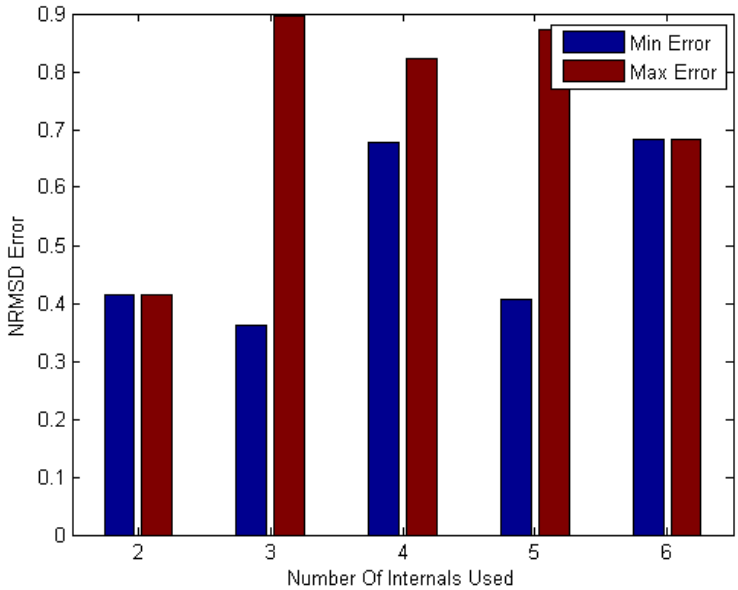


Figure 17 immediately afterwards shows the limits of the errors generated through the validation data sets through the maximum and minimum errors generated for the validation sets.

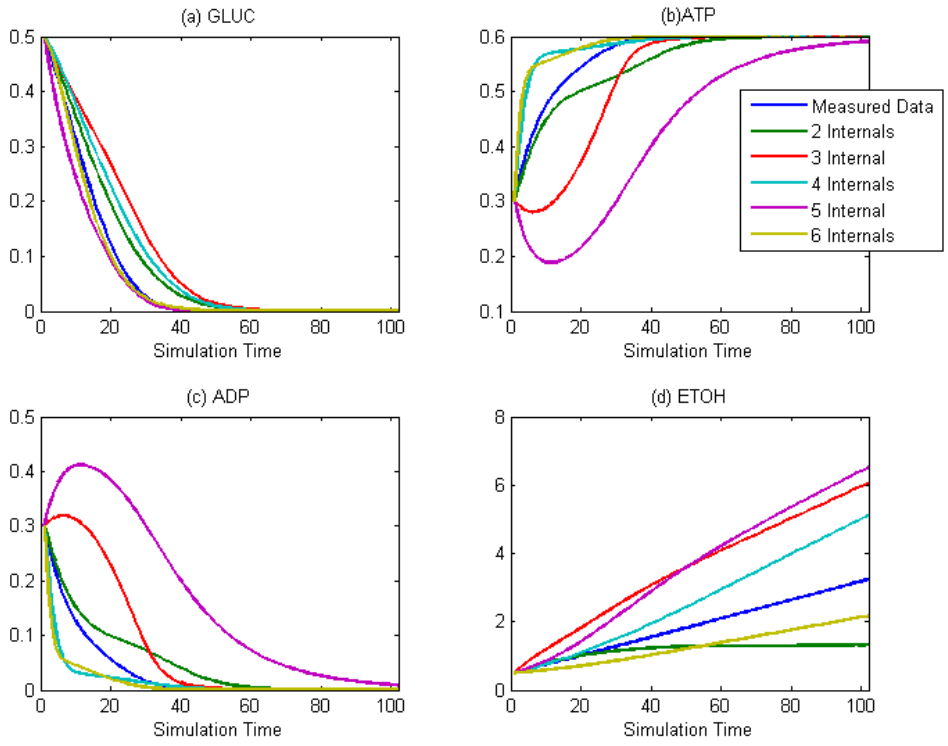


Figure 16: Metabolite Concentrations for Case 3 Externals (Validation Data)

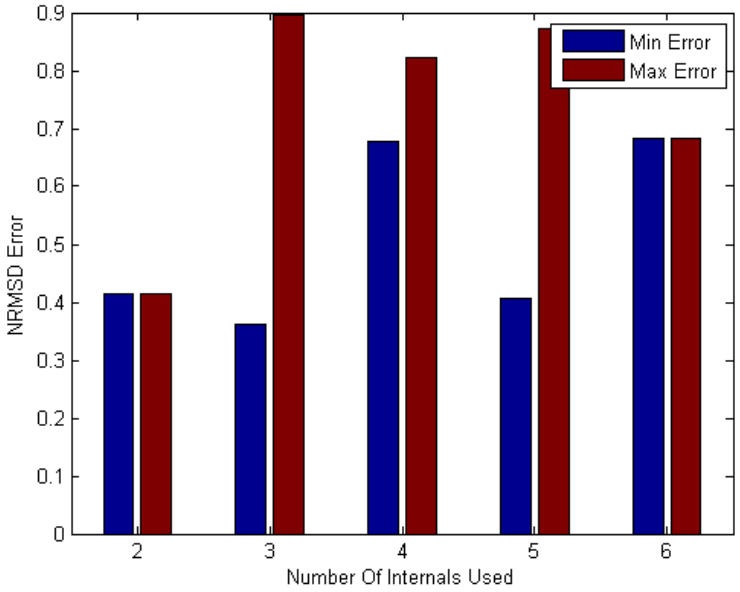


Figure 17: Error Limit Summary for Case 3 (Validation Sets)

4. Discussion

4.1 On Case Choices

The three test cases were chosen as they allowed us to explore various aspects of the abstraction process. Unlike the network in Case 1 which had four external metabolites and 2 internal metabolites, case 2 used only S and P as external metabolites, while some internal metabolites in case 3 were prohibited from elimination.

4.2 The Error Metric

The NRMSD error metric was used to report the error between per data point between the measured and model data. The advantage of this error metric over non-normalized RMSD is that NRMSD normalizes the error value to the range of values for each metabolite within a particular data set. Using absolute error values can be misleading. For example, a 0.02 difference in between a measurement and computed data point may seem small, when the reference metabolite concentration is 10, but staggering, when the metabolite concentration is a 2. Because we calculate the NRMSD for all boundary metabolites, where each can have a different range of concentrations, NRMSD seemed to be the most appropriate.

We also explored another error metric in attempt to force the convenience kinetic solution to more closely match the measured data earlier in the simulation time. We used a time-variant weighted NRMSD error, calculated by multipling the error of each point by the time left in the simulation over total simulation time. The front of the solutions was weighted more so than the steady state points. The quality of our solutiosn were not improved with this time-weighted NRMSD metric.

4.2 Initial Conditions

Initial conditions play a critical role in the ability of the algoirthm to idenitfy low-error white-box models. An internal metabolite with a non-zero initial concentration may influence the system signficnatly for the duration of the simulation. Thus, when internal metabolites have a zero initial concentration, or a concentration that does not significantly influence the state of the system, our approach yields reasonable results. Eliminating an internal metabolite with internal initial concentrations creates a model that cannot capture this effect. Eror values are thus high and continue to increase as additional metabolites are eliminated. The worst possibility for our modelling system occurs when external inputs are zero and internals are non-zero. In this case, the model with all internals eliminated fails to model the network because the network has no initial concentrations and thus no metabolites to turn over.

4.3 Activator/Inhibitor (A/I) Identification

Our algorithm can identify activation/inhibition activities for a given network. For case 2, all activators and inhibitors were known form published work. We were able to evaluate the accuracy of our algorithm in A/I identification. The best model generated for the initial solution, that with all internals included, used P once as an inhibitor once and S once as an activator. Additionally, five of the eight internal metabolites were used as activators for reactions. This was very similar to the original model and as S & P became not activators or inhibitors it caused the error to increase as in deviation from the original model. As our algorithm progresses and certain metabolites were eliminated, other metabolites, both internal and external, would change into

activators and inhibitors to make up for having other internals removed. This was expressly done with metabolite P which, as the algorithm iterated and the internal G and E nodes were removed, became an activator to perform the actions of the missing nodes. It is important to note that convenience kinetics cannot capture A/I relationships in isolation of product-reactant relationships. That is, the underlying stoichiometry must be determined prior to using convenience kinetics to explore A/I relationships in the system.

4.4 Relative Importance of Metabolites

For case 2, as shown in Figure 8, there is a large range of errors produced for each solution. This is caused by the metabolites used internally to model the proper network. In Table 3, it is seen that metabolites M1 and M2 almost always appear in the solution that generates the minimum error values, while they almost never appear in the solutions that generate the maximum error value. Figure 9 shows that solutions that include M1 or M2 never generate data with an error value greater than .21. These results highlight the importance of M1 and M2 to the model. There is a drastic difference between a model that doesn't include M1 or M2 to one that includes either or, ideally, both. M1 and M2 are both nodes directly in the pathway between the input and output. When the physical structure of the network is changed by eliminating one of the two nodes, the calculated data deviates from the measured data. When the second metabolite is eliminated, the data deviates even more. This is a direct consequence of removing an internal node. The results here thus go beyond sensitivity and perturbation analysis of parameters, and we are able to discover metabolite concentrations that play a more significant role in the network.

5. Summary and Conclusion

Identifying simplified kinetic expressions and using order-reduction methods promise to play a critical future role in creating genome-scale kinetic models. The work presented here utilizes genetic algorithms to explore the trade offs in expression accuracy and the inclusion of internal metabolite concentrations in these expressions. The approach essentially automatically infers new system co-dependencies when a metabolite is eliminated. Metabolites switch roles within the expressions and act as activators and inhibitors to compensate for missing variables. Our results for three test cases on validation data sets indicate that it is possible to eliminate some variables while modestly loosing accuracy. The ability to further understand such tradeoffs could lead to simplified data collection, and smaller expressions that can speed up simulations of larger systems.

REFERENCES

Altintas, M. M., C. K. Eddy, et al. (2006). "Kinetic modeling to optimize pentose fermentation in Zymomonas mobilis." Biotechnol Bioeng **94**(2): 273-95.

Chou, I. C. and E. O. Voit (2009). "Recent developments in parameter estimation and structure identification of biochemical and genomic systems." Math Biosci **219**(2): 57-83.
Costa, R. S., D. Machado, et al. (2011). "Critical perspective on the consequences of the limited availability of kinetic data in metabolic dynamic modelling." IET Syst Biol **5**(3): 157-63.
Dano, S., M. F. Madsen, et al. (2006). "Reduction of a biochemical model with preservation of its basic dynamic properties." FEBS J **273**(21): 4862-77.
Gonzalez, O. R., C. Kuper, et al. (2007). "Parameter estimation using simulated annealing for S-system models of biochemical networks." Bioinformatics **23**
Hatzimanikatis, V. and J. E. Bailey (1996). "MCA has more to say." J. Theor. Biol. **233**.
Ishii, N., Y. Suga, et al. (2007). "Dynamic simulation of an in vitro multi-enzyme system." FEBS Lett **581**(3): 413-20.
Laumanns, M., E. Zitzler, et al. (2001). On The Effects of Archiving, Elitism, and Density Based Selection in Evolutionairy Multi-Objective Optimization. Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization 181--196.
Liebermeister, W., U. Baur, et al. (2005). "Biochemical network models simplified by balanced truncation." FEBS J **272**(16): 4034-43.
Liebermeister, W. and E. Klipp (2006). "Bringing metabolic networks to life: convenience rate law and thermodynamic constraints." Theor Biol Med Model **3**: 41.
Mendes, P. and D. Kell (1998). "Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation." Bioinformatics **14**(10): 869-83.
Miller, B. L. and D. E. Goldberg (1995). "Genetic Algorithms, Tournament Selection, and the Effects of Noise." Complex Systems **9**: 193--212.
Moles, C. G., P. Mendes, et al. (2003). "Parameter estimation in biochemical pathways: a comparison of global optimization methods." Genome Res **13**(11): 2467-74.
Naval, P. C., L. G. Sison, et al. (2006). Metabolic network parameter inference using particle swarm optimization. International Conference on Molecular Systems Biology.
Nikerel, I. E., W. a. van Winden, et al. (2009). "Model reduction and a priori kinetic parameter identifiability analysis using metabolome time series for metabolic reaction networks with linlog kinetics." Metabolic engineering **11**: 20-30.
Okino, M. and M. Mavrovouniotis (1998). "Simplification of mathematical models of chemical reaction systems." Chem Rev: 391-408.
Peschel, M. and W. Mende (1986). Predator-Prey Model: Do We Live in a Volterra World?, Springer-Verlag.
Sridharan, G. V., S. Hassoun, et al. (2011). "Identification of biochemical network modules based on shortest retroactive distances." PLoS Comput Biol **7**(11): e1002262.
Visser, D. and J. J. Heijnen (2002). "The mathematics of metabolic control analysis revisited." Metab Eng **4**(2): 114-23.
Voit, E. O. (2000). Computational Analysis of Biochemical Systems. Cambridge University, Cambridge, UK.
Zuñiga, P. C., J. Pasia, et al. (2008). An ant colony optimization algorithm for parameter estimation and network inference problems in S-system models. International Conference on Molecular Systems Biology**:** 105.